Beginner›s Guide to LangChain - Building LLM Applications Made Easy

Data Science Dojo

Welcome to Data Science Dojo›s weekly newsletter, «The Data-Driven Dispatch».

Are you reliant on a single generative AI tool for all your tasks? Well, the chances are low.

The truth is that we are surrounded by tons of large language models. For some people, OpenAI›s GPT can work best for programming. But when it comes to writing, they might prefer Google›s Gemini.

Here›s the deal: People prefer different LLMs for different needs. This means the current LLM-powered tools do not provide a comprehensive solution.

Hence, there is a rising need to build more nuanced large language model applications than the current ones.

For example, tools that utilize different LLMs for different tasks, and understand data from any source and whatnot. The idea is to make an application become an expert in the domain it is made for.

But, how can one build AI applications like that?

That is where LangChain comes in!

What is LangChain?

LangChain is a popular and powerful tool that simplifies building LLM applications.

It provides an open-source orchestration framework that allows developers to build, integrate, and scale applications using large language models more efficiently. It comes in both Python and JavaScript Libraries. Read more

Why is it Important? Understanding LangChain Components

LangChain simplifies the development of LLM

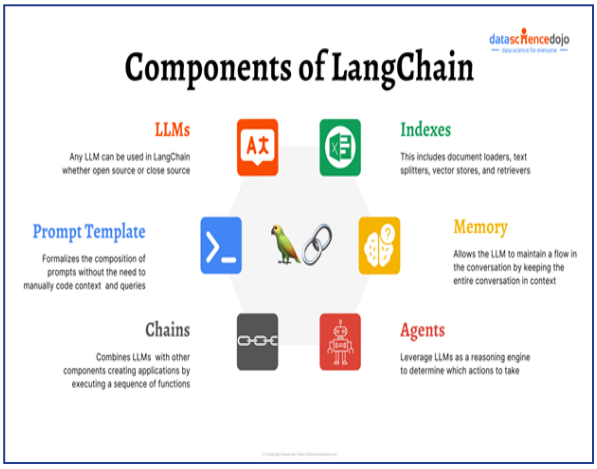
applications by employing abstractions.

These abstractions act like building blocks, representing common tasks and processes that are typically involved when working with language models.

By linking these blocks together–much like connecting pieces of a puzzle–developers can efficiently build applications.

The Benefit? This approach reduces the amount of coding needed to perform complex natural language processing (NLP) tasks making it way simpler.

We can call these building blocks components that can work together or by themselves, to facilitate the creation of powerful LLM applications.



Components of LangChain

Read: Learn the power of LangChain: A comprehensive guide to building custom Q&A chatbots

Use Cases of LangChain

LangChain opens the door to a lot of innovation with LLMs. It helps build GenAI applications that are versatile and cater to many functions. Here are some use cases of LangChain in LLM application development:

1. Chatbots
2. Summarization
3. Question Answering
4. Data Augmentation
5. Virtual Agents

LangChain Vs. LlamaIndex - Which Orchestration Framework is the Best?

Other than LangChain, LlamaIndex is widely used as an orchestration framework to simplify the process of building LLM applications.

Want to know which one works better? Here›s a comparison:

LlamaIndex Vs. LangChain datasciencedojo

Features	LlamaIndex	LangChain
Primary focus	Intelligent search, data indexing and retrieval	Building a wide range of Gen AI applications
Data handling	Ingesting, structuring, and accessing private or domain-specific data	Loading, processing, and indexing data for various uses
Customization	Offers tools for integrating private data into LLMs	Highly customizable, users can chain multiple components
Flexibility	Specialized for efficient and fast search	General-purpose framework with flexibility in application behavior
Deployment	Ideal for proprietary or specialized data	Facilitates the deployment of bespoke NLP applications
Use cases	Best for applications that require quick data lookup and retrieval	Suitable for applications that require complex interactions like chatbots, GQA, summarization

LlamaIndex Vs. LangChain

Read more LlamaIndex vs LangChain: Understand the key differences

Tutorial: How to Build Custom QnA Chatbots with LangChain, Step-by-Step

Curious to see LangChain in action? Here›s a comprehensive tutorial where Syed Hyder Ali Zaidi, Data Scientist and Data Science Dojo, provides a simplified explanation of LangChain, and its components.

He then builds a chatbot that not only understands natural language but also has the power to answer questions and provide insights from given CSV data.